
통합 구현

정기 수행 평가 07



Contents

1 프로젝트 생성

- 프로젝트 구조
- 구조 상세

2 화면 구현

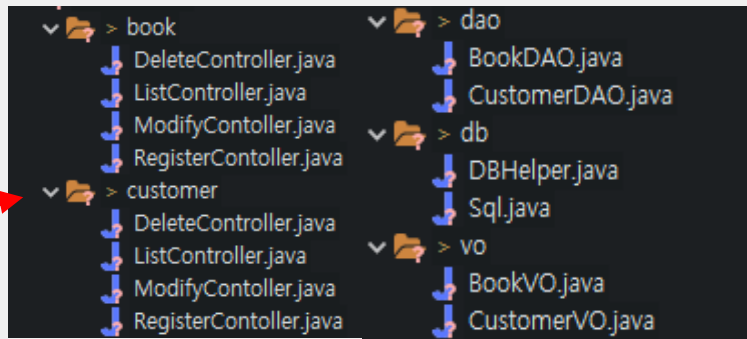
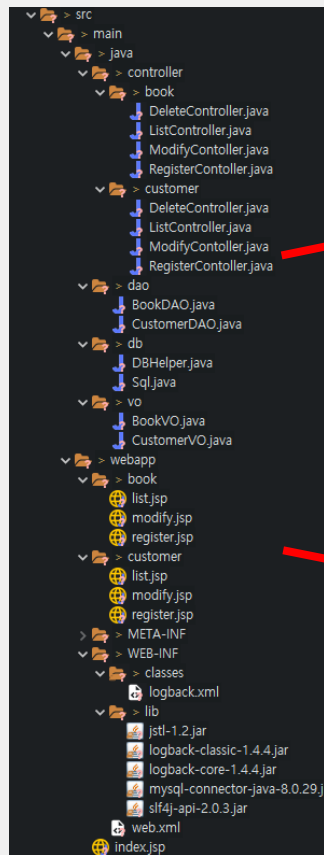
- 화면 구현

3 기능 구현

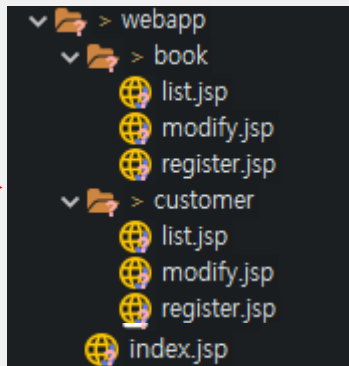
- Book 구현
- Customer 구현

1 프로젝트 생성

| 프로젝트 구조



컨트롤러 및 DB 관련 클래스



화면 구성

BookStore

[도서목록](#) [고객목록](#)

도서목록

[처음으로 도서등록](#)

도서번호	도서명	출판사	가격	관리
1	죽구의 역사	굿스포츠	7001	수정 삭제
2	죽구아는 여자	나무수	13000	수정 삭제
3	Hello World!	GA	89000	수정 삭제
9	올림픽 이야기	삼성당	7500	수정 삭제
10	Olympic Champions	Pearson	13000	수정 삭제

고객목록

[처음으로 고객등록](#)

고객번호	고객명	주소	휴대폰	관리
1	박지성	영국 맨체스터	000-5000-0002	수정 삭제
2	김연아	대한민국 서울	000-6000-0001	수정 삭제
3	장미란	대한민국 강원도	000-7000-0001	수정 삭제
4	추신수	미국 클리블랜드	000-8000-0001	수정 삭제

메인 화면

1 프로젝트 생성

| 구조 상세

```
public class DBHelper {  
    protected Connection conn = null;  
    protected PreparedStatement pstmt = null;  
    protected Statement stmt = null;  
    protected ResultSet rs = null;  
  
    public Connection getConnection() {  
        try {  
            DataSource ds = (DataSource)  
                new InitialContext().lookup  
                    ("java:comp/env/regularExam_java1_bookstore")  
  
            conn = ds.getConnection();  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
  
        return conn;  
    }  
  
    public void close() {  
        try {  
            if (rs != null) rs.close();  
            if (stmt != null) stmt.close();  
            if (pstmt != null) pstmt.close();  
            if (conn != null) conn.close();  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}  
//Java1_BookStore  
<Resource  
    name="regularExam_java1_bookstore"  
    auth="Container"  
    type="javax.sql.DataSource"  
    driverClassName="com.mysql.cj.jdbc.Driver"  
    url="jdbc:mysql://127.0.0.1:3306/java1_bookstore"  
    username="java1_book"  
    password="1234"  
    maxTotal="8"  
    maxIdle="8"  
    maxWaitMillis="3000"  
/>
```

```
package vo;  
  
public class CustomerVO {  
    private int custId;  
    private String custName;  
    private String custAddr;  
    private String custHp;  
  
    public int getCustId() {  
        return custId;  
    }  
  
    public void setCustId(int custId) {  
        this.custId = custId;  
    }  
}
```

```
import java.util.ArrayList;[]  
  
public class BookDAO extends DBHelper{  
    Logger logger = LoggerFactory.getLogger(this.getClass());  
  
    private static BookDAO instance = new BookDAO();  
    public static BookDAO getInstance() {  
        return instance;  
    }  
    private BookDAO() {}  
    // public BookVO selectBook(String bookId) {  
    logger.debug("selectBook Start...");  
    BookVO book = null;  
    try {  
        conn = getConnection();  
        pstmt = conn.prepareStatement(Sql.SELECT_BOOK);  
        pstmt.setString(1, bookId);  
        rs = pstmt.executeQuery();  
    }  
}
```

VO 및 DAO.java

```
package controller.book;  
  
import java.io.IOException;[]  
;  
@WebServlet("/book/delete.do")  
public class DeleteController extends HttpServlet{  
    private static final long serialVersionUID = 1L;  
  
    @Override  
    public void init() throws ServletException {}  
  
    @Override  
    protected void doGet(HttpServletRequest req, HttpServletResponse  
        String bookId = req.getParameter("bookId");  
        BookDAO.getInstance().deleteBook(bookId);  
        resp.sendRedirect("/Bookstore2/book/list.do");  
    }  
}
```

```
package db;  
  
public class Sql {  
    //Book  
    public static final String INSERT_BOOK = "INSERT INTO `book` VALUES(?,?,?,?)";  
    public static final String DELETE_BOOK = "DELETE FROM `book` WHERE `bookId`=?";  
    public static final String SELECT_BOOK = "SELECT * FROM `book` WHERE `bookId`=?";  
    public static final String SELECT_BOOKS = "SELECT * FROM `book`";  
    public static final String UPDATE_BOOK = "UPDATE `book` SET "  
        + "`bookName`=?, `publisher`=?, `price`=? "  
        + "WHERE `bookId`=?";  
  
    //Customer  
    public static final String INSERT_CUSTOMER = "INSERT INTO `customer` VALUES(?,?,?,?,?)";  
    public static final String DELETE_CUSTOMER = "DELETE FROM `customer` WHERE `custId`=?";  
    public static final String SELECT_CUSTOMER = "SELECT * FROM `customer` WHERE `custId`=?";  
    public static final String SELECT_CUSTOMERS = "SELECT * FROM `customer`";  
    public static final String UPDATE_CUSTOMER = "UPDATE `customer` SET "  
        + "`custName`=?, `custAddr`=?, `custHp`=? "  
        + "WHERE `custId`=?";  
}
```

2 화면 구현

| 화면 구현

BookStore

도서목록

[도서목록](#) [고객목록](#)

[처음으로 도서등록](#)

시작화면

도서번호	도서명	출판사	가격	관리
1	축구의 역사	굿스포츠	7001	수정 삭제
2	축구하는 여자	나무수	13000	수정 삭제
3	Hello World!	GA	89000	수정 삭제
9	올림픽 이야기	삼성당	7500	수정 삭제
10	Olympic Champions	Pearson	13000	수정 삭제

도서등록

[처음으로 도서목록](#)

도서번호	<input type="text"/>
도서명	<input type="text"/>
출판사	<input type="text"/>
가격	<input type="text"/>
<input type="button" value="등록"/>	

도서수정

[처음으로 도서목록](#)

도서번호	<input type="text" value="1"/>
도서명	<input type="text" value="축구의 역사"/>
출판사	<input type="text" value="굿스포츠"/>
가격	<input type="text" value="7001"/>
<input type="button" value="수정"/>	

Book 화면 구현

BookStore

고객목록

[도서목록](#) [고객목록](#)

[처음으로 고객등록](#)

시작화면

고객번호	고객명	주소	휴대폰	관리
1	박지성	영국 맨체스타	000-5000-0002	수정 삭제
2	김연아	대한민국 서울	000-6000-0001	수정 삭제
3	장미란	대한민국 강원도	000-7000-0001	수정 삭제
4	추신수	미국 클리블랜드	000-8000-0001	수정 삭제

고객등록

[처음으로 고객목록](#)

고객번호	<input type="text"/>
고객명	<input type="text"/>
주소	<input type="text"/>
휴대폰	<input type="text"/>
<input type="button" value="등록"/>	

고객수정

[처음으로 고객목록](#)

고객번호	<input type="text" value="1"/>
고객명	<input type="text" value="박지성"/>
주소	<input type="text" value="영국 맨체스타"/>
휴대폰	<input type="text" value="000-5000-0002"/>
<input type="button" value="수정"/>	

Customer 화면 구현

3 기능 구현

| Book 기능 구현

```
//Book
public static final String INSERT_BOOK = "INSERT INTO `book` VALUES(?,?,?)";
public static final String DELETE_BOOK = "DELETE FROM `book` WHERE `bookId`=?" ;
public static final String SELECT_BOOK = "SELECT * FROM `book` WHERE `bookId`=?" ;
public static final String SELECT_BOOKS = "SELECT * FROM `book`";
public static final String UPDATE_BOOK = "UPDATE `book` SET "
+ "`bookName`=?, `publisher`=?, `price`=?"
+ "WHERE `bookId`=?" ;
```

도서목록

처음으로 도서등록

도서번호	도서명	출판사	가격	관리
1	축구의 역사	굿스포츠	7001	수정 삭제
2	축구아는 여자	나무수	13000	수정 삭제
3	Hello World!	GA	89000	수정 삭제
9	올림픽 이야기	삼성당	7500	수정 삭제
10	Olympic Champions	Pearson	13000	수정 삭제

```
protected void doGet(HttpServletRequest req, HttpServletResponse res) {
    List<BookVO> books = BookDAO.getInstance().selectBooks();
    req.setAttribute("books", books);
}
```

```
public List<BookVO> selectBooks() {
    logger.debug("selectBooks Start...");
    List<BookVO> books = new ArrayList<>();
    try {
        conn = getConnection();
        stmt = conn.createStatement();
        rs = stmt.executeQuery(Sql.SELECT_BOOKS);

        while(rs.next()) {
            BookVO book = new BookVO();
            book.setBookId(rs.getInt(1));
            book.setBookName(rs.getString(2));
            book.setPublisher(rs.getString(3));
            book.setPrice(rs.getInt(4));
            books.add(book);
        }
        close();
    } catch (Exception e) {
        logger.error(e.getMessage());
    }
    logger.debug("selectBooks end...");
    return books;
}
```

3 기능 구현

| Book 기능 구현

```
//Book
public static final String INSERT_BOOK = "INSERT INTO `book` VALUES(?,?,?)";
public static final String DELETE_BOOK = "DELETE FROM `book` WHERE `bookId`=?" ;
public static final String SELECT_BOOK = "SELECT * FROM `book` WHERE `bookId`=?" ;
public static final String SELECT_BOOKS = "SELECT * FROM `book`";
public static final String UPDATE_BOOK = "UPDATE `book` SET "
    + "`bookName`=?, `publisher`=?, `price`=?"
    + "WHERE `bookId`=?" ;
```

도서목록

처음으로 도서등록

도서번호	도서명	출판사	가격	관리
1	축구의 역사	굿스포츠	7001	수정 삭제
2	축구아는 여자	나무수	13000	수정 삭제
3	Hello World!	GA	89000	수정 삭제
9	올림픽 이야기	삼성당	7500	수정 삭제
10	Olympic Champions	Pearson	13000	수정 삭제

도서목록

처음으로 도서등록

도서번호	도서명	출판사	가격	관리
1	축구의 역사	굿스포츠	7001	수정 삭제
2	축구아는 여자	나무수	13000	수정 삭제
3	Hello World!	GA	89000	수정 삭제
9	올림픽 이야기	삼성당	7500	수정 삭제
10	Olympic Champions	Pearson	13000	수정 삭제
12345	통합구현	김동근	12345	수정 삭제

도서등록

처음으로 도서등록

도서번호	12345
도서명	통합구현
출판사	김동근
가격	12345
<input type="button" value="등록"/>	

```
@WebServlet("/book/register.do")
public class RegisterController extends HttpServlet{
    private static final long serialVersionUID = 1L;

    @Override
    public void init() throws ServletException {}

    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse
        RequestDispatcher dispatcher =
            req.getRequestDispatcher("/book/register.jsp");
        dispatcher.forward(req, resp);
    }

    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse
```

```
String bookId = req.getParameter("bookId");
String bookName = req.getParameter("bookName");
String publisher = req.getParameter("publisher");
String price = req.getParameter("price");
```

```
BookVO book = new BookVO();
book.setBookId(bookId);
book.setBookName(bookName);
book.setPublisher(publisher);
book.setPrice(price);
```

```
BookDAO.getInstance().insertBook(book);
resp.sendRedirect("/Bookstore2/book/list.do");
}
```

```
public void insertBook(BookVO book) {
    logger.debug("insertBook Start...");
    try {
        conn = getConnection();
        pstmt = conn.prepareStatement(Sql.INSERT_BOOK);
        pstmt.setInt(1, book.getBookId());
        pstmt.setString(2, book.getBookName());
        pstmt.setString(3, book.getPublisher());
        pstmt.setInt(4, book.getPrice());
        pstmt.executeUpdate();

        close();
    } catch (Exception e) {
        logger.error(e.getMessage());
    }
    logger.debug("deleteBook end...");
}
```

Book 등록 구현

3 기능 구현

| Book 기능 구현

```
//Book
public static final String INSERT_BOOK = "INSERT INTO `book` VALUES(?,?,?,?)";
public static final String DELETE_BOOK = "DELETE FROM `book` WHERE `bookId`=?";
public static final String SELECT_BOOK = "SELECT * FROM `book` WHERE `bookId`=?";
public static final String SELECT_BOOKS = "SELECT * FROM `book`";
public static final String UPDATE_BOOK = "UPDATE `book` SET "
    + "`bookName`=?, `publisher`=?, `price`=?"
    + "WHERE `bookId`=?";
```

도서목록

처음으로 도서등록

도서번호	도서명	출판사	가격	관리
1	축구의 역사	굿스포츠	7001	수정 삭제
2	축구하는 여자	나무수	13000	수정 삭제
3	Hello World!	GA	89000	수정 삭제
9	올림픽 이야기	삼성당	7500	수정 삭제
10	Olympic Champions	Pearson	13000	수정 삭제
12345	통합구현	김동근	12345	수정 삭제

도서목록

처음으로 도서등록

도서번호	도서명	출판사	가격	관리
1	축구의 역사	굿스포츠	7001	수정 삭제
2	축구하는 여자	나무수	13000	수정 삭제
3	Hello World!	GA	89000	수정 삭제
9	올림픽 이야기	삼성당	7500	수정 삭제
10	Olympic Champions	Pearson	13000	수정 삭제
12345	통합구현	김동근	33333	수정 삭제

도서수정

처음으로 도서등록

도서번호	12345
도서명	통합구현
출판사	김동근
가격	33333
	<input type="button" value="수정"/>

```
@WebServlet("/book/modify.do")
public class ModifyController extends HttpServlet{
    private static final long serialVersionUID = 1L;

    @Override
    public void init() throws ServletException {}

    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse res) {
        String bookId = req.getParameter("bookId");
        BookVO book = BookDAO.getInstance().selectBook(bookId);
        req.setAttribute("book", book);

        RequestDispatcher dispatcher =
            req.getRequestDispatcher("/book/modify.jsp");
        dispatcher.forward(req, resp);
    }

    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse res) {
        String bookId = req.getParameter("bookId");
        String bookName = req.getParameter("bookName");
        String publisher = req.getParameter("publisher");
        String price = req.getParameter("price");

        BookVO book = new BookVO();
        book.setBookId(bookId);
        book.setBookName(bookName);
        book.setPublisher(publisher);
        book.setPrice(price);

        BookDAO.getInstance().updateBook(book);
        resp.sendRedirect("/Bookstore2/book/list.do");
    }
}

public BookVO selectBook(String bookId) {
    logger.debug("selectBook Start...");
    BookVO book = null;
    try {
        conn = getConnection();
        psmt = conn.prepareStatement(Sql.SELECT_BOOK);
        psmt.setString(1, bookId);
        rs = psmt.executeQuery();

        while(rs.next()) {
            book = new BookVO();
            book.setBookId(rs.getInt(1));
            book.setBookName(rs.getString(2));
            book.setPublisher(rs.getString(3));
            book.setPrice(rs.getInt(4));
        }
        close();
    } catch (Exception e) {
        logger.error(e.getMessage());
    }
    logger.debug("selectBook end...");
    return book;
}

public void updateBook(BookVO book) {
    logger.debug("updateBook Start...");
    try {
        conn = getConnection();
        psmt = conn.prepareStatement(Sql.UPDATE_BOOK);
        psmt.setString(1, book.getBookName());
        psmt.setString(2, book.getPublisher());
        psmt.setInt(3, book.getPrice());
        psmt.setInt(4, book.getBookId());
        psmt.executeUpdate();

        close();
    } catch (Exception e) {
        logger.error(e.getMessage());
    }
    logger.debug("updateBook end...");
}
```

Book 수정 구현

3 기능 구현

| Book 기능 구현

```
//Book
public static final String INSERT_BOOK = "INSERT INTO `book` VALUES(?,?,?,?)";
public static final String DELETE_BOOK = "DELETE FROM `book` WHERE `bookId`=?" ;
public static final String SELECT_BOOK = "SELECT * FROM `book` WHERE `bookId`=?" ;
public static final String SELECT_BOOKS = "SELECT * FROM `book`";
public static final String UPDATE_BOOK = "UPDATE `book` SET "
+ "`bookName`=?, `publisher`=?, `price`=?"
+ "WHERE `bookId`=?" ;
```

도서목록

처음으로 도서등록

도서번호	도서명	출판사	가격	관리
1	축구의 역사	굿스포츠	7001	수정 삭제
2	축구아는 여자	나무수	13000	수정 삭제
3	Hello World!	GA	89000	수정 삭제
9	올림픽 이야기	삼성당	7500	수정 삭제
10	Olympic Champions	Pearson	13000	수정 삭제
12345	통합구현	김동근	33333	수정 삭제

도서목록

처음으로 도서등록

도서번호	도서명	출판사	가격	관리
1	축구의 역사	굿스포츠	7001	수정 삭제
2	축구아는 여자	나무수	13000	수정 삭제
3	Hello World!	GA	89000	수정 삭제
10	Olympic Champions	Pearson	13000	수정 삭제

```
@WebServlet("/book/delete.do")
public class DeleteController extends HttpServlet{
    private static final long serialVersionUID = 1L;

    @Override
    public void init() throws ServletException {}

    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp) {
        String bookId = req.getParameter("bookId");
        BookDAO.getInstance().deleteBook(bookId);
        resp.sendRedirect("/Bookstore2/book/list.do");
    }

    public void deleteBook(String bookId) {
        logger.debug("deleteBook Start...");
        try {
            conn = getConnection();
            pstmt = conn.prepareStatement(Sql.DELETE_BOOK);
            pstmt.setString(1, bookId);
            pstmt.executeUpdate();

            close();
        } catch (Exception e) {
            logger.error(e.getMessage());
        }
        logger.debug("deleteBook end...");
    }
}
```

3 기능 구현

| customer 기능 구현

```
//Customer
public static final String INSERT_CUSTOMER = "INSERT INTO `customer` VALUES(?,?,?)";
public static final String DELETE_CUSTOMER = "DELETE FROM `customer` WHERE `custId`=?" ;
public static final String SELECT_CUSTOMER = "SELECT * FROM `customer` WHERE `custId`=?" ;
public static final String SELECT_CUSTOMERS = "SELECT * FROM `customer`";
public static final String UPDATE_CUSTOMER = "UPDATE `customer` SET "
+ "`custName`=?, `custAddr`=?, `custHp`=?"
+ "WHERE `custId`=?" ;
```

```
@WebServlet("/customer/list.do")
public class ListController extends HttpServlet{
    private static final long serialVersionUID = 1L;
    @Override
    public void init() throws ServletException {}

    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse res) {
        List<CustomerVO> customers =
            CustomerDAO.getInstance().selectCustomers();
        req.setAttribute("customers", customers);

        RequestDispatcher dispatcher =
            req.getRequestDispatcher("/customer/list.jsp");
        dispatcher.forward(req, res);
    }
}
```

고객목록

처음으로 고객등록

고객번호	고객명	주소	휴대폰	관리
1	박지성	영국 맨체스터	000-5000-0002	수정 삭제
2	김연아	대한민국 서울	000-6000-0001	수정 삭제
3	장미란	대한민국 강원도	000-7000-0001	수정 삭제
4	추신수	미국 클리블랜드	000-8000-0001	수정 삭제

```
public List<CustomerVO> selectCustomers() {
    logger.debug("selectCustomers Start...");
    List<CustomerVO> customers = new ArrayList<>();
    try {
        conn = getConnection();
        stmt = conn.createStatement();
        rs = stmt.executeQuery(Sql.SELECT_CUSTOMERS);

        while(rs.next()) {
            CustomerVO customer = new CustomerVO();
            customer.setCustId(rs.getInt(1));
            customer.setCustName(rs.getString(2));
            customer.setCustAddr(rs.getString(3));
            customer.setCustHp(rs.getString(4));
            customers.add(customer);
        }
        close();
    } catch (Exception e) {
        logger.error(e.getMessage());
    }
    logger.debug("selectCustomers end...");
    return customers;
}
```

3 기능 구현

| customer기능 구현

```
//Customer
public static final String INSERT_CUSTOMER = "INSERT INTO `customer` VALUES(?,?,?,?)";
public static final String DELETE_CUSTOMER = "DELETE FROM `customer` WHERE `custId`=?";
public static final String SELECT_CUSTOMER = "SELECT * FROM `customer` WHERE `custId`=?";
public static final String SELECT_CUSTOMERS = "SELECT * FROM `customer`";
public static final String UPDATE_CUSTOMER = "UPDATE `customer` SET "
+ "`custName`=?, `custAddr`=?, `custHp`=? "
+ "WHERE `custId`=?";
```

고객목록

처음으로 고객등록

고객번호	고객명	주소	휴대폰	관리
1	박지성	영국 맨체스터	000-5000-0002	수정 삭제
2	김연아	대한민국 서울	000-6000-0001	수정 삭제
3	장미란	대한민국 강원도	000-7000-0001	수정 삭제
4	추신수	미국 캘리포니아	000-8000-0001	수정 삭제

고객목록

처음으로 고객등록

고객번호	고객명	주소	휴대폰	관리
1	박지성	영국 맨체스터	000-5000-0002	수정 삭제
2	김연아	대한민국 서울	000-6000-0001	수정 삭제
3	장미란	대한민국 강원도	000-7000-0001	수정 삭제
4	추신수	미국 캘리포니아	000-8000-0001	수정 삭제
5	고객등록	고객등록	010-test-test	수정 삭제

고객등록

처음으로 고객등록

고객번호	5
고객명	고객등록
주소	고객등록
휴대폰	010-test-test
	등록

```
public void insertCustomer(CustomerVO customer) {
    logger.debug("insertCustomer Start...");
    try {
        conn = getConnection();
        pstmt = conn.prepareStatement(Sql.INSERT_CUSTOMER);
        pstmt.setInt(1, customer.getCustId());
        pstmt.setString(2, customer.getCustName());
        pstmt.setString(3, customer.getCustAddr());
        pstmt.setString(4, customer.getCustHp());
        pstmt.executeUpdate();

        close();
    } catch (Exception e) {
        logger.error(e.getMessage());
    }
    logger.debug("insertCustomer end...");
}
```

```
@WebServlet("/customer/register.do")
public class RegisterController extends HttpServlet{
    private static final long serialVersionUID = 1L;

    @Override
    public void init() throws ServletException {}

    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse res) {
        RequestDispatcher dispatcher =
            req.getRequestDispatcher("/customer/register.jsp");
        dispatcher.forward(req, res);
    }

    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse res) {
        String custId = req.getParameter("custId");
        String custName = req.getParameter("custName");
        String custAddr = req.getParameter("custAddr");
        String custHp = req.getParameter("custHp");

        CustomerVO customer = new CustomerVO();
        customer.setCustId(custId);
        customer.setCustName(custName);
        customer.setCustAddr(custAddr);
        customer.setCustHp(custHp);

        CustomerDAO.getInstance().insertCustomer(customer);
        resp.sendRedirect("/Bookstore2/customer/list.do");
    }
}
```

customer 등록 구현

3 기능 구현

| customer 기능 구현

```
//Customer
public static final String INSERT_CUSTOMER = "INSERT INTO `customer` VALUES(?,?,?,?)";
public static final String DELETE_CUSTOMER = "DELETE FROM `customer` WHERE `custId`=?";
public static final String SELECT_CUSTOMER = "SELECT * FROM `customer` WHERE `custId`=?";
public static final String SELECT_CUSTOMERS = "SELECT * FROM `customer`";
public static final String UPDATE_CUSTOMER = "UPDATE `customer` SET "
+ "`custName`=?, `custAddr`=?, `custHp`=? "
+ "WHERE `custId`=?";
```

고객목록

처음으로 고객등록

고객번호	고객명	주소	휴대폰	관리
1	박지성	영국 맨체스터	000-5000-0002	수정 삭제
2	김연아	대한민국 서울	000-6000-0001	수정 삭제
3	장미란	대한민국 강원도	000-7000-0001	수정 삭제
4	추신수	미국 클리블랜드	000-8000-0001	수정 삭제
5	고객등록	고객등록	010-test-test	수정 삭제

고객목록

처음으로 고객등록

고객번호	고객명	주소	휴대폰	관리
1	박지성	영국 맨체스터	000-5000-0002	수정 삭제
2	김연아	대한민국 서울	000-6000-0001	수정 삭제
3	장미란	대한민국 강원도	000-7000-0001	수정 삭제
4	추신수	미국 클리블랜드	000-8000-0001	수정 삭제
5	고객수정	고객수정	010-test-test	수정 삭제

고객수정

처음으로 고객등록

고객번호	5
고객명	고객수정
주소	고객수정
휴대폰	010-test-test
	수정

```
public CustomerVO selectCustomer(String custId) {
    logger.debug("selectCustomer Start...");
    CustomerVO customer = null;
    try {
        conn = getConnection();
        pstmt = conn.prepareStatement(Sql.SELECT_CUSTOMER);
        pstmt.setString(1, custId);
        rs = pstmt.executeQuery();

        while(rs.next()) {
            customer = new CustomerVO();
            customer.setCustId(rs.getInt(1));
            customer.setCustName(rs.getString(2));
            customer.setCustAddr(rs.getString(3));
            customer.setCustHp(rs.getString(4));
        }
        close();
    } catch (Exception e) {
        logger.error(e.getMessage());
    }
    logger.debug("selectCustomer end...");
    return customer;
}
```

```
public void updateCustomer(CustomerVO customer) {
    logger.debug("updateCustomer Start...");
    try {
        conn = getConnection();
        pstmt = conn.prepareStatement(Sql.UPDATE_CUSTOMER);
        pstmt.setString(1, customer.getCustName());
        pstmt.setString(2, customer.getCustAddr());
        pstmt.setString(3, customer.getCustHp());
        pstmt.setInt(4, customer.getCustId());
        pstmt.executeUpdate();

        close();
    } catch (Exception e) {
        logger.error(e.getMessage());
    }
    logger.debug("updateCustomer end...");
}
```

```
@WebServlet("/customer/modify.do")
public class ModifyController extends HttpServlet{
    private static final long serialVersionUID = 1L;

    @Override
    public void init() throws ServletException {}

    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp) {
        String custId = req.getParameter("custId");
        CustomerVO customer =
            CustomerDAO.getInstance().selectCustomer(custId);
        req.setAttribute("customer", customer);

        RequestDispatcher dispatcher =
            req.getRequestDispatcher("/customer/modify.jsp");
        dispatcher.forward(req, resp);
    }
}
```

```
@Override
protected void doPost(HttpServletRequest req, HttpServletResponse resp) {
    String custId = req.getParameter("custId");
    String custName = req.getParameter("custName");
    String custAddr = req.getParameter("custAddr");
    String custHp = req.getParameter("custHp");

    CustomerVO customer = new CustomerVO();
    customer.setCustId(custId);
    customer.setCustName(custName);
    customer.setCustAddr(custAddr);
    customer.setCustHp(custHp);

    CustomerDAO.getInstance().updateCustomer(customer);
    resp.sendRedirect("/Bookstore2/customer/list.do");
}
```

customer 수정 구현

3 기능 구현

| customer 기능 구현

```
//Customer
public static final String INSERT_CUSTOMER = "INSERT INTO `customer` VALUES(?,?,?)";
public static final String DELETE_CUSTOMER = "DELETE FROM `customer` WHERE `custId`=?" ;
public static final String SELECT_CUSTOMER = "SELECT * FROM `customer` WHERE `custId`=?" ;
public static final String SELECT_CUSTOMERS = "SELECT * FROM `customer`";
public static final String UPDATE_CUSTOMER = "UPDATE `customer` SET "
+ "`custName`=?, `custAddr`=?, `custHp`=?"
+ "WHERE `custId`=?" ;
```

고객목록

처음으로 고객등록

고객번호	고객명	주소	휴대폰	관리
1	박지성	영국 맨체스터	000-5000-0002	수정 삭제
2	김연아	대한민국 서울	000-6000-0001	수정 삭제
3	장미란	대한민국 강원도	000-7000-0001	수정 삭제
4	추신수	미국 클리블랜드	000-8000-0001	수정 삭제
5	고객수정	고객수정	010-test-test	수정 삭제

고객목록

처음으로 고객등록

고객번호	고객명	주소	휴대폰	관리
1	박지성	영국 맨체스터	000-5000-0002	수정 삭제
2	김연아	대한민국 서울	000-6000-0001	수정 삭제
4	추신수	미국 클리블랜드	000-8000-0001	수정 삭제

```
public void deleteCustomer(String custId) {
    logger.debug("deleteCustomer Start...");
    try {
        conn = getConnection();
        pstmt = conn.prepareStatement(Sql.DELETE_CUSTOMER);
        pstmt.setString(1, custId);
        pstmt.executeUpdate();

        close();
    } catch (Exception e) {
        logger.error(e.getMessage());
    }
    logger.debug("deleteCustomer end...");
}
```

```
@WebServlet("/customer/delete.do")
public class DeleteController extends HttpServlet{
    private static final long serialVersionUID = 1L;

    @Override
    public void init() throws ServletException {}

    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
        String custId = req.getParameter("custId");
        CustomerDAO.getInstance().deleteCustomer(custId);
        resp.sendRedirect("/Bookstore2/customer/list.do");
    }
}
```

정기 수행 평가

07.통합 구현 Bookstore



Thank you!

Kim Dong Geun
From GreenAcademy
2022.11.18